



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: ☒ The ACM Digital Library ☐ The Guide

control system termination request tentative continue job cont

SEARCH



[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used

control system termination request tentative continue job control

Found 81,306 of 151,219

Sort results
by

relevance



[Save results to a Binder](#)

[Try an Advanced Search](#)

Display
results

expanded form



[Search Tips](#)

Try this search in [The ACM Guide](#)

☐ Open results in a new
window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Distributed operating systems](#)

Andrew S. Tanenbaum, Robbert Van Renesse

December 1985 **ACM Computing Surveys (CSUR)**, Volume 17 Issue 4

Full text available: [pdf\(5.49 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Distributed operating systems have many aspects in common with centralized ones, but they also differ in certain ways. This paper is intended as an introduction to distributed operating systems, and especially to current university research about them. After a discussion of what constitutes a distributed operating system and how it is distinguished from a computer network, various key design issues are discussed. Then several examples of current research projects are examined in some detail ...

2 [Programming languages for distributed computing systems](#)

Henri E. Bal, Jennifer G. Steiner, Andrew S. Tanenbaum

September 1989 **ACM Computing Surveys (CSUR)**, Volume 21 Issue 3

Full text available: [pdf\(6.50 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

When distributed systems first appeared, they were programmed in traditional sequential languages, usually with the addition of a few library procedures for sending and receiving messages. As distributed applications became more commonplace and more sophisticated, this ad hoc approach became less satisfactory. Researchers all over the world began designing new programming languages specifically for implementing distributed applications. These languages and their history, their underlying pr ...

3 [Energy-aware systems: Energy-efficient, utility accrual scheduling under resource constraints for mobile embedded systems](#)

Haisang Wu, Binoy Ravindran, E. Douglas Jensen, Peng Li

September 2004 **Proceedings of the fourth ACM international conference on Embedded software**

Full text available: [pdf\(379.20 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We present an energy-efficient real-time scheduling algorithm called the *Resource-constrained Energy-Efficient Utility Accrual Algorithm* (or ReUA). ReUA considers an application model where activities are subject to time/utility function-time constraints, resource dependencies including mutual exclusion constraints, and statistical performance

requirements including probabilistically satisfied, activity (timeliness) utility bounds. Further, ReUA targets mobile embedded systems where syste ...

Keywords: energy-efficient scheduling, real-time systems, time/utility functions, utility accrual scheduling

4 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**

Full text available:  [pdf\(4.21 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

5 A Survey of Techniques for Synchronization and Recovery in Decentralized Computer Systems

Walter H. Kohler


June 1981 **ACM Computing Surveys (CSUR)**, Volume 13 Issue 2

Full text available:  [pdf\(3.33 MB\)](#) Additional Information: [full citation](#), [references](#), [citing](#), [index terms](#)

6 4.2BSD and 4.3BSD as examples of the UNIX system

John S. Quarterman, Abraham Silberschatz, James L. Peterson

December 1985 **ACM Computing Surveys (CSUR)**, Volume 17 Issue 4


Full text available:  [pdf\(4.07 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citing](#), [index terms](#), [review](#)

This paper presents an in-depth examination of the 4.2 Berkeley Software Distribution, Virtual VAX-11 Version (4.2BSD), which is a version of the UNIX Time-Sharing System. There are notes throughout on 4.3BSD, the forthcoming system from the University of California at Berkeley. We trace the historical development of the UNIX system from its conception in 1969 until today, and describe the design principles that have guided this development. We then present the internal data structures and ...

7 Draft Proposed: American National Standard—Graphical Kernel System

Technical Committee X3H3 - Computer Graphics


February 1984 **ACM SIGGRAPH Computer Graphics**, Volume 18 Issue SI

Full text available:  [pdf\(16.07 MB\)](#) Additional Information: [full citation](#)

8 An Unclever Time-Sharing System

Caxton C. Foster

January 1971 **ACM Computing Surveys (CSUR)**, Volume 3 Issue 1

Full text available:  [pdf\(1.85 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citing](#), [index terms](#)


This paper describes the internal structure of a time-sharing system in some detail. This system is dedicated to providing remote access, and has a simple file structure. It is

intended for use in a university type environment where there are many short jobs that will profit from one- or two-second turnaround. Despite its simplicity, this system can serve as a useful introduction to the problems encountered by the designers of any time-sharing system. Included are a discussion of the comman ...

9 Data base directions: the next steps

John L. Berg

November 1976 , Volume 8 , 8 Issue 4 , 2

Full text available:  [pdf\(9.95 MB\)](#)

Additional Information: [full citation](#), [abstract](#)


What information about data base technology does a manager need to make prudent decisions about using this new technology? To provide this information the National Bureau of Standards and the Association for Computing Machinery established a workshop of approximately 80 experts in five major subject areas. The five subject areas were auditing, evolving technology, government regulations, standards, and user experience. Each area prepared a report contained in these proceedings. The proceedings p ...

Keywords: DBMS, auditing, cost/benefit analysis, data base, data base management, government regulation, management objectives, privacy, security, standards, technology assessment, user experience

10 Human-computer interface development: concepts and systems for its management

H. Rex Hartson, Deborah Hix

March 1989 **ACM Computing Surveys (CSUR)**, Volume 21 Issue 1

Full text available:  [pdf\(7.97 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Human-computer interface management, from a computer science viewpoint, focuses on the process of developing quality human-computer interfaces, including their representation, design, implementation, execution, evaluation, and maintenance. This survey presents important concepts of interface management: dialogue independence, structural modeling, representation, interactive tools, rapid prototyping, development methodologies, and control structures. *Dialogue independence* is th ...

11 The Integrated Dictionary/Directory System

Frank W. Allen, Mary E. S. Loomis, Michael V. Mannino

June 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 2

Full text available:  [pdf\(2.71 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

12 A programming environment for a timeshared system

Richard P. Gabriel, Martin E. Frost

April 1984 **Proceedings of the first ACM SIGSOFT/SIGPLAN software engineering symposium on Practical software development environments**, Volume 19 , 9 Issue 5 , 3

Full text available:  [pdf\(859.14 KB\)](#)


Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In 1968 the Stanford Artificial Intelligence Laboratory began to construct a programming environment from a PDP-10, a pre-TOPS-10 DEC1 timesharing system, and some innovative terminal hardware. By now this has developed into a programming environment for a KL-10 that integrates our editor with various other system functions, especially the Lisp subsystem. We use the term 'SAIL' to refer to the Stanford A. I. Lab KL-10 computer running the WAITS timesharing system. [Ha ...

13 A Survey of Some Theoretical Aspects of Multiprocessing

J. L. Baer

January 1973 **ACM Computing Surveys (CSUR)**, Volume 5 Issue 1

Full text available:  pdf(4.05 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

14 The command interpreter and command language design of the com-share
COMMANDER II system

Steven S. Muchnick

October 1976 **Proceedings of the annual conference**

Full text available:  pdf(515.98 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The command language of the COMMANDER II time-sharing system was designed on the basis of a well-developed philosophy of the facilities a time-sharing utility should provide its users and the form in which the two should communicate. The philosophy includes concern for symmetry and completeness of the facilities provided, simplicity of input formats, confirmation of potentially disastrous effects, full use of the capabilities of the terminal, compatibility between batch and interactive use, ...

15 Database concurrency control using data flow graphs

M. H. Eich, David L. Wells

June 1988 **ACM Transactions on Database Systems (TODS)**, Volume 13 Issue 2

Full text available:  pdf(2.42 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

A specialized data flow graph, Database Flow Graph (DBFG) is introduced. DBFGs may be used for scheduling database operations, particularly in an MIMD database machine environment. A DBFG explicitly maintains intertransaction and intratransaction dependencies, and is constructed from the Transaction Flow Graphs (TFG) of active transactions. A TFG, in turn, is the generalization of a query tree used, for example, in DIRECT [15]. All DBFG schedules ...

16 Curriculum recommendations for graduate professional programs in information
systems

May 1972 **Communications of the ACM**, Volume 15 Issue 5

Full text available:  pdf(4.00 MB)

Additional Information: [full citation](#), [references](#), [citations](#)

Keywords: education, information analysis, information systems development, management information systems, management systems, system design, systems analysis

17 System R: relational approach to database management

M. M. Astrahan, M. W. Blasgen, D. D. Chamberlin, K. P. Eswaran, J. N. Gray, P. P. Griffiths, W. F. King, R. A. Lorie, P. R. McJones, J. W. Mehl, G. R. Putzolu, I. L. Traiger, B. W. Wade, V. Watson

June 1976 **ACM Transactions on Database Systems (TODS)**, Volume 1 Issue 2

Full text available:  pdf(3.18 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

System R is a database management system which provides a high level relational data interface. The systems provides a high level of data independence by isolating the end user as much as possible from underlying storage structures. The system permits definition of a

variety of relational views on common underlying data. Data control features are provided, including authorization, integrity assertions, triggered transactions, a logging and recovery subsystem, and facilities for maintaining ...

Keywords: authorization, data structures, database, index structures, locking, nonprocedural language, recovery, relational model

18 Document Formatting Systems: Survey, Concepts, and Issues

Richard Furuta, Jeffrey Scofield, Alan Shaw

September 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 3

Full text available:  pdf(5.36 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)



19 Network Protocols

Andrew S. Tanenbaum

December 1981 **ACM Computing Surveys (CSUR)**, Volume 13 Issue 4

Full text available:  pdf(3.37 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)



20 M65MP: An experiment in OS/360 multiprocessing

Bernard I. Witt

January 1968 **Proceedings of the 1968 23rd ACM national conference**

Full text available:  pdf(1.07 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)



The System/360 Model 65 multiprocessing system (M65MP) described in this paper exists and is in operation at the IBM facility in Gaithersburg, Maryland. The Gaithersburg programming effort is the base for IBM's support of multiprocessing announced on January 3, 1968. The only purpose of this paper, however, is to relate strictly personal observations about the development period and the results accomplished.

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)



Terms used

test control system termination request tentative continue job control

Found 79,411 of 151,219

Sort results
by

relevance



[Save results to a Binder](#)

[Try an Advanced Search](#)

[Try this search in The ACM Guide](#)

Display
results

expanded form



[Search Tips](#)

☐ Open results in a new
window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)


Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Fast detection of communication patterns in distributed executions](#)

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**

Full text available:  pdf(4.21 MB)


Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

2 [Programming languages for distributed computing systems](#)

Henri E. Bal, Jennifer G. Steiner, Andrew S. Tanenbaum

September 1989 **ACM Computing Surveys (CSUR)**, Volume 21 Issue 3

Full text available:  pdf(6.50 MB)


Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

When distributed systems first appeared, they were programmed in traditional sequential languages, usually with the addition of a few library procedures for sending and receiving messages. As distributed applications became more commonplace and more sophisticated, this ad hoc approach became less satisfactory. Researchers all over the world began designing new programming languages specifically for implementing distributed applications. These languages and their history, their underlying pr ...

3 [Data base directions: the next steps](#)

John L. Berg

November 1976 , Volume 8 , 8 Issue 4 , 2

Full text available:  pdf(9.95 MB)

Additional Information: [full citation](#), [abstract](#)


What information about data base technology does a manager need to make prudent decisions about using this new technology? To provide this information the National Bureau of Standards and the Association for Computing Machinery established a workshop of approximately 80 experts in five major subject areas. The five subject areas were auditing, evolving technology, government regulations, standards, and user experience. Each area prepared a report contained in these proceedings. The proceedings p ...

Keywords: DBMS, auditing, cost/benefit analysis, data base, data base management, government regulation, management objectives, privacy, security, standards, technology assessment, user experience

4 Distributed operating systems

Andrew S. Tanenbaum, Robbert Van Renesse

December 1985 **ACM Computing Surveys (CSUR)**, Volume 17 Issue 4

Full text available:  pdf(5.49 MB)


Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Distributed operating systems have many aspects in common with centralized ones, but they also differ in certain ways. This paper is intended as an introduction to distributed operating systems, and especially to current university research about them. After a discussion of what constitutes a distributed operating system and how it is distinguished from a computer network, various key design issues are discussed. Then several examples of current research projects are examined in some detail ...

5 Human-computer interface development: concepts and systems for its management

H. Rex Hartson, Deborah Hix

March 1989 **ACM Computing Surveys (CSUR)**, Volume 21 Issue 1

Full text available:  pdf(7.97 MB)


Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Human-computer interface management, from a computer science viewpoint, focuses on the process of developing quality human-computer interfaces, including their representation, design, implementation, execution, evaluation, and maintenance. This survey presents important concepts of interface management: dialogue independence, structural modeling, representation, interactive tools, rapid prototyping, development methodologies, and control structures. *Dialogue independence* is th ...

6 A Survey of Techniques for Synchronization and Recovery in Decentralized Computer Systems

Walter H. Kohler

June 1981 **ACM Computing Surveys (CSUR)**, Volume 13 Issue 2


Full text available:  pdf(3.33 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

7 A Survey of Some Theoretical Aspects of Multiprocessing

J. L. Baer


January 1973 **ACM Computing Surveys (CSUR)**, Volume 5 Issue 1

Full text available:  pdf(4.05 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

8 Curriculum recommendations for graduate professional programs in information systems

May 1972 **Communications of the ACM**, Volume 15 Issue 5

Full text available:  pdf(4.00 MB)

Additional Information: [full citation](#), [references](#), [citations](#)

Keywords: education, information analysis, information systems development, management information systems, management systems, system design, systems analysis

9 The Integrated Dictionary/Directory System

Frank W. Allen, Mary E. S. Loomis, Michael V. Mannino


June 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 2

Full text available:  pdf(2.71 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

10 4.2BSD and 4.3BSD as examples of the UNIX system

John S. Quarterman, Abraham Silberschatz, James L. Peterson

December 1985 **ACM Computing Surveys (CSUR)**, Volume 17 Issue 4

Full text available:  pdf(4.07 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This paper presents an in-depth examination of the 4.2 Berkeley Software Distribution, Virtual VAX-11 Version (4.2BSD), which is a version of the UNIX Time-Sharing System. There are notes throughout on 4.3BSD, the forthcoming system from the University of California at Berkeley. We trace the historical development of the UNIX system from its conception in 1969 until today, and describe the design principles that have guided this development. We then present the internal data structures and ...

11 Reliability Issues in Computing System Design

B. Randell, P. Lee, P. C. Treleaven


June 1978 **ACM Computing Surveys (CSUR)**, Volume 10 Issue 2

Full text available:  pdf(3.95 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

12 Draft Proposed: American National Standard—Graphical Kernel System

Technical Committee X3H3 - Computer Graphics


February 1984 **ACM SIGGRAPH Computer Graphics**, Volume 18 Issue SI

Full text available:  pdf(16.07 MB) Additional Information: [full citation](#)

13 An Unclever Time-Sharing System

Caxton C. Foster

January 1971 **ACM Computing Surveys (CSUR)**, Volume 3 Issue 1

Full text available:  pdf(1.85 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper describes the internal structure of a time-sharing system in some detail. This system is dedicated to providing remote access, and has a simple file structure. It is intended for use in a university type environment where there are many short jobs that will profit from one- or two-second turnaround. Despite its simplicity, this system can serve as a useful introduction to the problems encountered by the designers of any time-sharing system. Included are a discussion of the command ...

14 Level II technical support in a distributed computing environment

Tim Leehane

September 1996 **Proceedings of the 24th annual ACM SIGUCCS conference on User services**

Full text available:  pdf(5.73 MB) Additional Information: [full citation](#), [references](#), [index terms](#)

15 Office Information Systems and Computer Science

Clarence A. Ellis, Gary J. Nutt

January 1980 **ACM Computing Surveys (CSUR)**, Volume 12 Issue 1

Full text available:  pdf(2.87 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

16 Document Formatting Systems: Survey, Concepts, and Issues

Richard Furuta, Jeffrey Scofield, Alan Shaw


September 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 3

Full text available:  pdf(5.36 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

17 A programming environment for a timeshared system

Richard P. Gabriel, Martin E. Frost

April 1984 **Proceedings of the first ACM SIGSOFT/SIGPLAN software engineering symposium on Practical software development environments**, Volume 19 , 9 Issue 5 , 3

Full text available:  pdf(859.14 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In 1968 the Stanford Artificial Intelligence Laboratory began to construct a programming environment from a PDP-10, a pre-TOPS-10 DEC1 timesharing system, and some innovative terminal hardware. By now this has developed into a programming environment for a KL-10 that integrates our editor with various other system functions, especially the Lisp subsystem. We use the term 'SAIL' to refer to the Stanford A. I. Lab KL-10 computer running the WAITS timesharing system. [Ha ...

18 The computer assisted software engineering (CASE) system

William Scott Amey

September 1979 **Proceedings of the 4th international conference on Software engineering**

Full text available:  pdf(398.18 KB) Additional Information: [full citation](#), [abstract](#), [index terms](#)

The CASE system provides a comprehensive software development tool that aids engineers, designers, programmers, and managers. The CASE system guides its users through the tasks of documentation, design, coding, testing, configuration control, and status analysis.

19 Micropipelines

I. E. Sutherland

June 1989 **Communications of the ACM**, Volume 32 Issue 6

Full text available:  pdf(3.16 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)


The pipeline processor is a common paradigm for very high speed computing machinery. Pipeline processors provide high speed because their separate stages can operate concurrently, much as different people on a manufacturing assembly line work concurrently on material passing down the line. Although the concurrency of pipeline processors makes their design a demanding task, they can be found in graphics processors, in signal processing devices, in integrated circuit components for doing arit ...

20 System R: relational approach to database management

M. M. Astrahan, M. W. Blasgen, D. D. Chamberlin, K. P. Eswaran, J. N. Gray, P. P. Griffiths, W. F. King, R. A. Lorie, P. R. McJones, J. W. Mehl, G. R. Putzolu, I. L. Traiger, B. W. Wade, V.

Watson

June 1976 **ACM Transactions on Database Systems (TODS)**, Volume 1 Issue 2

Full text available:  [pdf\(3.18 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


System R is a database management system which provides a high level relational data interface. The system provides a high level of data independence by isolating the end user as much as possible from underlying storage structures. The system permits definition of a variety of relational views on common underlying data. Data control features are provided, including authorization, integrity assertions, triggered transactions, a logging and recovery subsystem, and facilities for maintaining ...

Keywords: authorization, data structures, database, index structures, locking, nonprocedural language, recovery, relational model

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)